Data/Album.java

package com.example.spotifyapp.Data;  
  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.util.Log;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.util.ArrayList;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
  
public class Album {  
 private String albumName;  
 private String imageUrl;  
 private String releaseDate;  
 private int totalTrack;  
 private String albumId;  
 private ArrayList<String> artistNames=new ArrayList<>();  
 private ArrayList<Track> tracks=new ArrayList<>();  
 public static ExecutorService *networkExecutorService* = Executors.*newFixedThreadPool*(4);  
 private Bitmap bitmapImage=null;  
  
 public Album() {  
 this.albumName = "";  
 this.imageUrl = "";  
 this.releaseDate = "";  
 this.totalTrack = 0;  
 this.albumId = "";  
 this.artistNames=new ArrayList<>();  
 }  
  
 public Album(String albumName, String imageUrl, String releaseDate, int totalTrack, String albumId, ArrayList<String> artistNames) {  
 this.albumName = albumName;  
 this.imageUrl = imageUrl;  
 this.releaseDate = releaseDate;  
 this.totalTrack = totalTrack;  
 this.albumId = albumId;  
 this.artistNames = artistNames;  
 }  
  
 public String getAlbumName() {  
 return albumName;  
 }  
  
 public Bitmap getImageBitmap() {  
 convertImagetoBitmap();  
 while(!*networkExecutorService*.isTerminated()){  
  
 }  
 System.*out*.println("Fetching Finished");  
 *networkExecutorService*=Executors.*newFixedThreadPool*(4);  
 return this.bitmapImage;  
 }  
  
 public void convertImagetoBitmap(){  
 *networkExecutorService*.execute(new Runnable() {  
 @Override  
 public void run() {  
 try {  
 URL url = new URL(imageUrl);  
 Bitmap image = BitmapFactory.*decodeStream*(url.openConnection().getInputStream());  
 bitmapImage=image;  
 } catch(IOException e) {  
 System.*out*.println(e);  
 }finally {  
 *networkExecutorService*.shutdown();  
 }  
 }  
 });  
 }  
  
 public String getReleaseDate() {  
 return releaseDate;  
 }  
  
 public int getTotalTrack() {  
 return totalTrack;  
 }  
  
 public String getAlbumId() {  
 return albumId;  
 }  
  
 public ArrayList<String> getArtistNames() {  
 return artistNames;  
 }  
  
 public ArrayList<Track> getTracks() {  
 return tracks;  
 }  
  
 public Album(String albumName, String imageUrl, String releaseDate, int totalTrack, String albumId, ArrayList<String> artistNames, ArrayList<Track> tracks) {  
 this.albumName = albumName;  
 this.imageUrl = imageUrl;  
 this.releaseDate = releaseDate;  
 this.totalTrack = totalTrack;  
 this.albumId = albumId;  
 this.artistNames = artistNames;  
 this.tracks = tracks;  
 }  
}

/data/token.java

package com.example.spotifyapp.Data;  
  
public class Token {  
 private String token;  
 private long expireIn;  
 private long timeGetToken;  
  
 public Token() {  
 token="";  
 expireIn=0;  
 timeGetToken=0;  
 }  
  
 public Token(String token, long expireIn, long timeGetToken) {  
 this.token = token;  
 this.expireIn = expireIn;  
 this.timeGetToken = timeGetToken;  
 }  
  
 public boolean isEmpty(){  
 return token=="";  
 }  
  
 public String getToken() {  
 return token;  
 }  
  
 public void setToken(String token) {  
 this.token = token;  
 }  
  
 public long getExpireIn() {  
 return expireIn;  
 }  
  
 public void setExpireIn(long expireIn) {  
 this.expireIn = expireIn;  
 }  
  
 public long getTimeGetToken() {  
 return timeGetToken;  
 }  
  
 public void setTimeGetToken(long timeGetToken) {  
 this.timeGetToken = timeGetToken;  
 }  
}

data/track.java

package com.example.spotifyapp.Data;  
  
public class Track {  
 private String id;  
 private int trackNumber;  
 private String trackName;  
 private long trackDuration;  
 private String trackPreviewUrl;  
  
 public Track(String id, int trackNumber, String trackName, long trackDuration, String trackPreviewUrl) {  
 this.id = id;  
 this.trackNumber = trackNumber;  
 this.trackName = trackName;  
 this.trackDuration = trackDuration;  
 this.trackPreviewUrl = trackPreviewUrl;  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public int getTrackNumber() {  
 return trackNumber;  
 }  
  
 public String getTrackName() {  
 return trackName;  
 }  
  
 public long getTrackDuration() {  
 return trackDuration;  
 }  
  
 public String getTrackPreviewUrl() {  
 return trackPreviewUrl;  
 }  
}

db/appDatabase

package com.example.spotifyapp.db;  
  
import android.content.Context;  
  
import androidx.room.Database;  
import androidx.room.Room;  
import androidx.room.RoomDatabase;  
  
@Database(entities = {TrackFavourite.class}, version = 1)  
public abstract class AppDatabase extends RoomDatabase {  
 public abstract TrackDao trackDao();  
  
 private static AppDatabase *INSTANCE*;  
 public static AppDatabase getDbInstance(Context context){  
 if(*INSTANCE* == null){  
 *INSTANCE* = Room.*databaseBuilder*(context.getApplicationContext(), AppDatabase.class, "DB\_NAME").allowMainThreadQueries().build();  
 }  
 return *INSTANCE*;  
 }  
}

db/TrackDao

package com.example.spotifyapp.db;  
  
import androidx.room.Dao;  
import androidx.room.Delete;  
import androidx.room.Insert;  
import androidx.room.Query;  
  
import java.util.List;  
  
@Dao  
public interface TrackDao {  
 @Query("SELECT \* FROM TrackFavourite")  
 List<TrackFavourite> getAllTrackIds();  
  
 @Insert  
 void insertTrack(TrackFavourite... trackFavourites);  
  
 @Delete  
 void delete(TrackFavourite trackFavourite);  
}

db/trackFavourite

package com.example.spotifyapp.db;  
  
import androidx.room.ColumnInfo;  
import androidx.room.Entity;  
import androidx.room.PrimaryKey;  
  
@Entity  
public class TrackFavourite {  
 @PrimaryKey(autoGenerate = true)  
 public int uid;  
  
 @ColumnInfo(name = "track\_id")  
 public String trackId;  
}

Service/JsonService

package com.example.spotifyapp.Service;  
  
import com.example.spotifyapp.Data.Album;  
import com.example.spotifyapp.Data.Token;  
import com.example.spotifyapp.Data.Track;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.IOException;  
import java.util.ArrayList;  
  
public class JsonService {  
 private ArrayList<Album> albums = new ArrayList<>();  
 private ArrayList<Track> tracks = new ArrayList<>();  
 private MusicDataService service = new MusicDataService();  
 private Token tokenObj;  
  
 public Token extractToken(String json){  
 try{  
 JSONObject jsonObject = new JSONObject(json);  
 String token = jsonObject.getString("access\_token");  
 long expires\_in = jsonObject.getLong("expires\_in");  
 long now = System.*currentTimeMillis*();  
 tokenObj= new Token(token,expires\_in,now);  
 return tokenObj;  
 }catch (JSONException e){  
 e.printStackTrace();  
 }  
 return null;  
 }  
  
 public ArrayList<Album> extractNewRelease(String json) {  
 try {  
 albums.clear();  
 JSONObject jsonAlbumObject = new JSONObject(json);  
 JSONObject jsonAlbum = jsonAlbumObject.getJSONObject("albums");  
 JSONArray albumArray = jsonAlbum.getJSONArray("items");  
 for (int i = 0; i < albumArray.length(); i++) {  
 String albumName = albumArray.getJSONObject(i).getString("name");  
 String imageUrl = albumArray.getJSONObject(i).getJSONArray("images").getJSONObject(0).getString("url");  
 String releaseDate = albumArray.getJSONObject(i).getString("release\_date");  
 int totalTrack = albumArray.getJSONObject(i).getInt("total\_tracks");  
 JSONArray artistArray = albumArray.getJSONObject(i).getJSONArray("artists");  
 ArrayList<String> artists = new ArrayList<>();  
 for (int y = 0; y < artistArray.length(); y++) {  
 String artistName = artistArray.getJSONObject(y).getString("name");  
 artists.add(artistName);  
 }  
 String albumId = albumArray.getJSONObject(i).getString("id");  
 Album album = new Album(albumName, imageUrl, releaseDate, totalTrack, albumId, artists);  
 albums.add(album);  
 }  
 return albums;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
  
 public Album extractAlbumById(String json) {  
 try {  
 tracks.clear();  
 JSONObject albumById = new JSONObject(json);  
 String albumName = albumById.getString("name");  
 String imageUrl = albumById.getJSONArray("images").getJSONObject(0).getString("url");  
 String releaseDate = albumById.getString("release\_date");  
 String albumId = albumById.getString("id");  
 int totalTrack = albumById.getInt("total\_tracks");  
 JSONArray artistArray = albumById.getJSONArray("artists");  
 ArrayList<String> artists = new ArrayList<>();  
 for (int i = 0; i < artistArray.length(); i++) {  
 String artistName = artistArray.getJSONObject(i).getString("name");  
 artists.add(artistName);  
 }  
 JSONArray tracksArray = albumById.getJSONObject("tracks").getJSONArray("items");  
  
 for (int i = 0; i < tracksArray.length(); i++) {  
 String trackId = tracksArray.getJSONObject(i).getString("id");  
 int trackNumber = tracksArray.getJSONObject(i).getInt("track\_number");  
 String trackName = tracksArray.getJSONObject(i).getString("name");  
 long duringMilisecond = tracksArray.getJSONObject(i).getLong("duration\_ms");  
 String previewUrl = tracksArray.getJSONObject(i).getString("preview\_url");  
 Track track = new Track(trackId, trackNumber, trackName, duringMilisecond, previewUrl);  
 tracks.add(track);  
 }  
 Album album = new Album(albumName, imageUrl, releaseDate, totalTrack, albumId, artists, tracks);  
 return album;  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
  
 public ArrayList<Track> extractFavourite(String json){  
 ArrayList<Track> favouriteTracks = new ArrayList<>();  
 try{  
  
 JSONObject favourite = new JSONObject(json);  
 JSONArray tracksArray = favourite.getJSONArray("tracks");  
  
 for (int i = 0; i < tracksArray.length(); i++) {  
 String trackId = tracksArray.getJSONObject(i).getString("id");  
 int trackNumber = tracksArray.getJSONObject(i).getInt("track\_number");  
 String trackName = tracksArray.getJSONObject(i).getString("name");  
 long duringMilisecond = tracksArray.getJSONObject(i).getLong("duration\_ms");  
 String previewUrl = tracksArray.getJSONObject(i).getString("preview\_url");  
 Track track = new Track(trackId, trackNumber, trackName, duringMilisecond, previewUrl);  
 favouriteTracks.add(track);  
 }  
 return favouriteTracks;  
 }catch (JSONException e){  
 e.printStackTrace();  
 }  
 return null;  
 }  
}

Service/MusicDataService

package com.example.spotifyapp.Service;  
  
import android.os.Handler;  
import android.os.Looper;  
  
import com.example.spotifyapp.Data.Token;  
import com.example.spotifyapp.R;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.net.URLEncoder;  
import java.util.Base64;  
import java.util.LinkedHashMap;  
import java.util.Map;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
  
public class MusicDataService{  
 public static ExecutorService *networkExecutorService* = Executors.*newFixedThreadPool*(4);  
 private String responseString="";  
  
 public String getResponseString(){  
 return responseString;  
 }  
  
 public void getNewRelease(String token){  
 connectNewRelease(token);  
 while(!*networkExecutorService*.isTerminated()){  
  
 }  
 System.*out*.println("Fetching Finished");  
 *networkExecutorService*=Executors.*newFixedThreadPool*(4);  
 }  
  
 public void getToken(){  
 connectToken();  
 while(!*networkExecutorService*.isTerminated()){  
  
 }  
 System.*out*.println("Fetching Finished");  
 *networkExecutorService*=Executors.*newFixedThreadPool*(4);  
 }  
  
 public void getAlbumById(String albumId, String token){  
 connectGetAlbumById(albumId, token);  
 while(!*networkExecutorService*.isTerminated()){  
  
 }  
 System.*out*.println("Fetching Finished");  
 *networkExecutorService*=Executors.*newFixedThreadPool*(4);  
 }  
  
 public void connectGetAlbumById(String albumId, String token){  
 *networkExecutorService*.execute(new Runnable() {  
 @Override  
 public void run() {  
 HttpURLConnection connection = null;  
 BufferedReader reader;  
 String line;  
 StringBuffer response = new StringBuffer();  
 try{  
 URL url = new URL("https://api.spotify.com/v1/albums/"+albumId);  
 connection = (HttpURLConnection) url.openConnection();  
 connection.setRequestMethod("GET");  
 connection.setConnectTimeout(5000);  
 connection.setReadTimeout(5000);  
 connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");  
 connection.setRequestProperty ("Authorization", "Bearer "+token);  
  
 int status = connection.getResponseCode();  
  
 if(status != 200){  
 reader = new BufferedReader(new InputStreamReader(connection.getErrorStream()));  
 while((line = reader.readLine()) !=null){  
 response.append(line);  
 }  
 reader.close();  
 }else{  
 reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 while((line = reader.readLine())!=null){  
 response.append(line);  
 }  
 reader.close();  
 }  
 System.*out*.println(response.toString());  
 responseString=response.toString();  
 }catch(IOException e){  
 e.printStackTrace();  
 }finally{  
 connection.disconnect();  
 *networkExecutorService*.shutdown();  
 }  
 }  
 });  
 }  
  
  
 public void connectNewRelease(String token){  
 *networkExecutorService*.execute(new Runnable() {  
 @Override  
 public void run() {  
 System.*out*.println("TOKEN: "+ token);  
 HttpURLConnection connection = null;  
 BufferedReader reader;  
 String line;  
 StringBuffer response = new StringBuffer();  
 try {  
 URL url = new URL("https://api.spotify.com/v1/browse/new-releases");  
 connection = (HttpURLConnection) url.openConnection();  
 connection.setRequestMethod("GET");  
 connection.setConnectTimeout(5000);  
 connection.setReadTimeout(5000);  
 connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");  
 connection.setRequestProperty("Authorization", "Bearer " + token);  
  
 int status = connection.getResponseCode();  
  
 if (status != 200) {  
 reader = new BufferedReader(new InputStreamReader(connection.getErrorStream()));  
 while ((line = reader.readLine()) != null) {  
 response.append(line);  
 }  
 reader.close();  
 } else {  
 reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 while ((line = reader.readLine()) != null) {  
 response.append(line);  
 }  
 reader.close();  
 }  
 responseString=response.toString();  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 connection.disconnect();  
 networkExecutorService.shutdown();  
 }  
 }  
 });  
 }  
  
  
 public void connectToken(){  
 networkExecutorService.execute(new Runnable() {  
 @Override  
 public void run() {  
 HttpURLConnection connection = null;  
 BufferedReader reader;  
 String line;  
 StringBuffer response = new StringBuffer();  
 try{  
 String auth="3fa081d1f2aa4d839c71bd4858fa7e52:9b846cadfb6e46648747627d54dcd449";  
 String basicAuth = "Basic " + Base64.getEncoder().encodeToString(auth.getBytes());  
  
 Map<String, Object> params = new LinkedHashMap<>();  
 params.put("grant\_type", "client\_credentials");  
  
 StringBuilder postData = new StringBuilder();  
 for(Map.Entry<String, Object> param : params.entrySet()){  
 if(postData.length() != 0) postData.append('&');  
 postData.append(URLEncoder.encode(param.getKey(), "UTF-8"));  
 postData.append('=');  
 postData.append(URLEncoder.encode(String.valueOf(param.getValue()),"UTF-8"));  
 }  
 byte[] postDataBytes = postData.toString().getBytes("UTF-8");  
  
 URL url = new URL("https://accounts.spotify.com/api/token");  
 connection = (HttpURLConnection) url.openConnection();  
 connection.setRequestMethod("POST");  
 connection.setConnectTimeout(5000);  
 connection.setReadTimeout(5000);  
 connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");  
 connection.setRequestProperty ("Authorization", basicAuth);  
 connection.setDoOutput(true);  
 connection.getOutputStream().write(postDataBytes);  
  
 int status = connection.getResponseCode();  
  
 if(status != 200){  
 reader = new BufferedReader(new InputStreamReader(connection.getErrorStream()));  
 while((line = reader.readLine()) !=null){  
 response.append(line);  
 }  
 reader.close();  
 }else{  
 reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 while((line = reader.readLine())!=null){  
 response.append(line);  
 }  
 reader.close();  
 }  
 responseString=response.toString();  
 }catch(IOException e){  
 e.printStackTrace();  
 }finally{  
 connection.disconnect();  
 networkExecutorService.shutdown();  
 }  
 }  
 });  
 }  
  
 public void getFavourite(String params, String token){  
 connectFavourite(params, token);  
 while(!networkExecutorService.isTerminated()){  
  
 }  
 System.out.println("Fetching Finished");  
 networkExecutorService=Executors.newFixedThreadPool(4);  
 }  
  
 public void connectFavourite(String params, String token){  
 networkExecutorService.execute(new Runnable() {  
 @Override  
 public void run() {  
 HttpURLConnection connection = null;  
 BufferedReader reader;  
 String line;  
 StringBuffer response = new StringBuffer();  
 try{  
 URL url = new URL("https://api.spotify.com/v1/tracks?ids="+params);  
 connection = (HttpURLConnection) url.openConnection();  
 connection.setRequestMethod("GET");  
 connection.setConnectTimeout(5000);  
 connection.setReadTimeout(5000);  
 connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");  
 connection.setRequestProperty ("Authorization", "Bearer "+token);  
  
 int status = connection.getResponseCode();  
  
 if(status != 200){  
 reader = new BufferedReader(new InputStreamReader(connection.getErrorStream()));  
 while((line = reader.readLine()) !=null){  
 response.append(line);  
 }  
 reader.close();  
 }else{  
 reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 while((line = reader.readLine())!=null){  
 response.append(line);  
 }  
 reader.close();  
 }  
 System.out.println(response.toString());  
 responseString = response.toString();  
 }catch(IOException e){  
 e.printStackTrace();  
 }finally{  
 connection.disconnect();  
 networkExecutorService.shutdown();  
 }  
 }  
 });  
 }  
  
}

AlbumActivity.java

package com.example.spotifyapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.media.MediaPlayer;  
import android.os.Bundle;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.spotifyapp.Data.Album;  
import com.example.spotifyapp.Data.Token;  
import com.example.spotifyapp.Data.Track;  
import com.example.spotifyapp.Service.JsonService;  
import com.example.spotifyapp.Service.MusicDataService;  
import com.example.spotifyapp.db.AppDatabase;  
import com.example.spotifyapp.db.TrackFavourite;  
  
import java.io.IOException;  
import java.util.ArrayList;  
  
public class AlbumActivity extends AppCompatActivity {  
 private String albumId="";  
 private Token tokenObj = new Token();  
 private JsonService jsonService = new JsonService();  
 private MusicDataService mService = new MusicDataService();  
 private Album album = new Album();  
 private RecyclerView mRecyclerView;  
 private MediaPlayer mediaPlayer=new MediaPlayer();  
 private String oldTrackName="";  
 private TextView albumName, releaseDate, trackCount;  
 private ImageView imageAlbum;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_album*);  
 if(this.getIntent().getExtras()!=null){  
 albumId=this.getIntent().getExtras().getString("albumId");  
 }  
 getAlbumById(albumId);  
 setAlbumAttribute(album);  
  
 mRecyclerView = findViewById(R.id.*trackRecycleView*);  
 mRecyclerView.setHasFixedSize(true);  
 AlbumAdapter albumAdapter = new AlbumAdapter(this, album.getTracks());  
 mRecyclerView.setLayoutManager(new LinearLayoutManager(this));  
 mRecyclerView.setAdapter(albumAdapter);  
  
 albumAdapter.setOnItemClickListener(new AlbumAdapter.OnItemClickListener(){  
 @Override  
 public void onItemClicked(Track track) {  
  
 }  
  
 @Override  
 public void playButtonClicked(Track track) {  
 if(!mediaPlayer.isPlaying()){  
 prepareMedia(track);  
 mediaPlayer.start();  
 }  
 Toast.*makeText*(AlbumActivity.this, track.getTrackName()+" is playing", Toast.*LENGTH\_LONG*).show();  
 oldTrackName = track.getTrackName();  
 }  
  
 @Override  
 public void stopButtonClicked(Track track) {  
 if(mediaPlayer.isPlaying()){  
 Toast.*makeText*(AlbumActivity.this, oldTrackName+" is stopped", Toast.*LENGTH\_LONG*).show();  
 mediaPlayer.stop();  
 mediaPlayer.reset();  
 mediaPlayer.release();  
 mediaPlayer = new MediaPlayer();  
 }  
 }  
  
 @Override  
 public void addFavoriteClicked(Track track) {  
 AppDatabase db = AppDatabase.*getDbInstance*(AlbumActivity.this.getApplicationContext());  
 TrackFavourite trackFavourite = new TrackFavourite();  
 trackFavourite.trackId = track.getId();  
 db.trackDao().insertTrack(trackFavourite);  
 Toast.*makeText*(AlbumActivity.this, track.getTrackName()+" added to playlist", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
  
 }  
  
 public void setAlbumAttribute(Album album){  
 albumName = findViewById(R.id.*albumName*);  
 releaseDate = findViewById(R.id.*releaseDate*);  
 trackCount = findViewById(R.id.*trackCount*);  
 imageAlbum = findViewById(R.id.*albumImage*);  
 albumName.setText(album.getAlbumName());  
 releaseDate.setText(album.getReleaseDate());  
 trackCount.setText(String.*valueOf*(album.getTotalTrack()));  
 imageAlbum.setImageBitmap(album.getImageBitmap());  
 }  
  
 public void getAlbumById(String albumId){  
 checkExpireToken();  
 mService.getAlbumById(albumId, tokenObj.getToken());  
 album = jsonService.extractAlbumById(mService.getResponseString());  
 }  
  
  
 public void getNewToken(){  
 mService.getToken();  
 tokenObj=jsonService.extractToken(mService.getResponseString());  
 }  
  
 public void checkExpireToken(){  
 long now = System.*currentTimeMillis*();  
 if ((now - tokenObj.getTimeGetToken()) > tokenObj.getExpireIn() || tokenObj.isEmpty()) {  
 getNewToken();  
 }  
 }  
  
 public void prepareMedia(Track track){  
 try{  
 mediaPlayer.setDataSource(track.getTrackPreviewUrl());  
 mediaPlayer.prepare();  
 }catch (IOException e){  
 e.printStackTrace();  
 }  
 }  
}

AlbumAdapter.java

package com.example.spotifyapp;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.spotifyapp.Data.Album;  
import com.example.spotifyapp.Data.Track;  
  
import java.text.DecimalFormat;  
import java.util.ArrayList;  
import java.util.concurrent.TimeUnit;  
  
public class AlbumAdapter extends RecyclerView.Adapter<AlbumAdapter.MyViewHolder> {  
 ArrayList<Track> tracks=new ArrayList<>();  
 Context context;  
 private OnItemClickListener listener;  
  
 public interface OnItemClickListener {  
 void onItemClicked(Track track);  
 void playButtonClicked(Track track);  
 void stopButtonClicked(Track track);  
 void addFavoriteClicked(Track track);  
 }  
  
 public void setOnItemClickListener(OnItemClickListener listener){  
 this.listener =listener;  
 }  
  
 public AlbumAdapter(Context context, ArrayList<Track> tracks) {  
 this.context= context;  
 this.tracks=tracks;  
 }  
  
 @NonNull  
 @Override  
 public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 LayoutInflater inflater = LayoutInflater.*from*(context);  
 View view = inflater.inflate(R.layout.*track\_item*, parent, false);  
 return new MyViewHolder(view, listener);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {  
 DecimalFormat df = new DecimalFormat("##");  
 String temp = tracks.get(position).getTrackNumber()+" "+  
 tracks.get(position).getTrackName()+" - "+(TimeUnit.*MILLISECONDS*.toMinutes(tracks.get(position).getTrackDuration()))+  
 ":"+String.*format*("%02d", (TimeUnit.*MILLISECONDS*.toSeconds(tracks.get(position).getTrackDuration())/60));  
 holder.trackDetail.setText(temp);  
 }  
  
 @Override  
 public int getItemCount() {  
 return tracks.size();  
 }  
  
 public class MyViewHolder extends RecyclerView.ViewHolder{  
 TextView trackDetail;  
 Button playButton, stopButton;  
 ImageView addFavorite;  
  
 public MyViewHolder(@NonNull View itemView, final OnItemClickListener listener) {  
 super(itemView);  
 trackDetail = itemView.findViewById(R.id.*trackDetail*);  
 playButton = itemView.findViewById(R.id.*playButton*);  
 stopButton = itemView.findViewById(R.id.*stopButton*);  
 addFavorite = itemView.findViewById(R.id.*addFavorite*);  
  
 itemView.setOnClickListener((v)->{  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.*NO\_POSITION*){  
 listener.onItemClicked(tracks.get(position));  
 }  
 }  
 });  
  
 playButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.*NO\_POSITION*){  
 listener.playButtonClicked(tracks.get(position));  
 }  
 }  
 }  
 });  
  
 stopButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.*NO\_POSITION*){  
 listener.stopButtonClicked(tracks.get(position));  
 }  
 }  
 }  
 });  
  
 addFavorite.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.*NO\_POSITION*){  
 listener.addFavoriteClicked(tracks.get(position));  
 }  
 }  
 }  
 });  
 }  
  
  
 }  
}

ArtistAdapter

package com.example.spotifyapp;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
public class ArtistAdapter extends RecyclerView.Adapter<ArtistAdapter.MyViewHolder> {  
 ArrayList<String> artists=new ArrayList<>();  
 Context context;  
  
 public ArtistAdapter(Context context, ArrayList<String> artists){  
 this.context=context;  
 this.artists=artists;  
 }  
  
 @NonNull  
 @Override  
 public ArtistAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 LayoutInflater inflater = LayoutInflater.*from*(context);  
 View view = inflater.inflate(R.layout.*artist\_item*, parent, false);  
 return new ArtistAdapter.MyViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ArtistAdapter.MyViewHolder holder, int position) {  
 holder.artistNameView.setText(artists.get(position));  
 }  
  
 @Override  
 public int getItemCount() {  
 return artists.size();  
 }  
  
 public class MyViewHolder extends RecyclerView.ViewHolder{  
 TextView artistNameView;  
 public MyViewHolder(@NonNull View itemView) {  
 super(itemView);  
 artistNameView = itemView.findViewById(R.id.*artistName*);  
 }  
 }  
}

FavouriteListActivity.java

package com.example.spotifyapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.media.MediaPlayer;  
import android.os.Bundle;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.spotifyapp.Data.Token;  
import com.example.spotifyapp.Data.Track;  
import com.example.spotifyapp.Service.JsonService;  
import com.example.spotifyapp.Service.MusicDataService;  
import com.example.spotifyapp.db.AppDatabase;  
import com.example.spotifyapp.db.TrackFavourite;  
  
import java.io.IOException;  
import java.sql.Array;  
import java.util.ArrayList;  
import java.util.List;  
  
public class FavoriteListActivity extends AppCompatActivity {  
 private ArrayList<String> favoriteString=new ArrayList<>();  
 private Token tokenObj = new Token();  
 private JsonService jsonService = new JsonService();  
 private MusicDataService mService = new MusicDataService();  
 private ArrayList<Track> tracks=new ArrayList<>();  
 private RecyclerView mRecyclerView;  
 private MediaPlayer mediaPlayer=new MediaPlayer();  
 private String oldTrackName="";  
 List<TrackFavourite> trackFavourites=new ArrayList<>();  
 FavouriteAdapter favouriteAdapter;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_favorite\_list*);  
 initView();  
  
  
 }  
  
  
 void initView(){  
 loadTrackFavourite();  
 TextView status=findViewById(R.id.*status*);  
 if(favoriteString.size()>0){  
 status.setText("");  
 String params=getParamsFavourite(favoriteString);  
 getFavourite(params);  
 mRecyclerView = findViewById(R.id.*favoriteListRecycleView*);  
 mRecyclerView.setHasFixedSize(true);  
 favouriteAdapter = new FavouriteAdapter(this, tracks);  
 mRecyclerView.setLayoutManager(new LinearLayoutManager(this));  
 mRecyclerView.setAdapter(favouriteAdapter);  
 favouriteAdapter.setOnItemClickListener(new FavouriteAdapter.OnItemClickListener(){  
 @Override  
 public void onItemClicked(Track track) {  
  
 }  
  
 @Override  
 public void playButtonClicked(Track track) {  
 if(!mediaPlayer.isPlaying()){  
 prepareMedia(track);  
 mediaPlayer.start();  
 }  
 Toast.*makeText*(FavoriteListActivity.this, track.getTrackName()+" is playing", Toast.*LENGTH\_LONG*).show();  
 oldTrackName = track.getTrackName();  
 }  
  
 @Override  
 public void stopButtonClicked(Track track) {  
 if(mediaPlayer.isPlaying()){  
 Toast.*makeText*(FavoriteListActivity.this, oldTrackName+" is stopped", Toast.*LENGTH\_LONG*).show();  
 mediaPlayer.stop();  
 mediaPlayer.reset();  
 mediaPlayer.release();  
 mediaPlayer = new MediaPlayer();  
 }  
 }  
  
 @Override  
 public void addFavoriteClicked(Track track) {  
 AppDatabase db = AppDatabase.*getDbInstance*(FavoriteListActivity.this.getApplicationContext());  
 int index = -1;  
 for(int i=0;i<trackFavourites.size();i++){  
 if(track.getId().contains(trackFavourites.get(i).trackId)){  
 index = i;  
 break;  
 }  
 }  
 db.trackDao().delete(trackFavourites.get(index));  
 Toast.*makeText*(FavoriteListActivity.this, track.getTrackName()+" Deleted from playlist", Toast.*LENGTH\_LONG*).show();  
 initView();  
 }  
 });  
 }else{  
 status.setText("Empty List");  
 }  
 }  
  
 public void prepareMedia(Track track){  
 try{  
 mediaPlayer.setDataSource(track.getTrackPreviewUrl());  
 mediaPlayer.prepare();  
 }catch (IOException e){  
 e.printStackTrace();  
 }  
 }  
  
 private String getParamsFavourite(ArrayList<String> arrayList){  
 String params="";  
 for(int i=0;i<arrayList.size();i++){  
 params+=arrayList.get(i)+",";  
 }  
 params=params.substring(0, params.length()-1);  
 return params;  
 }  
  
 private void loadTrackFavourite(){  
 trackFavourites.clear();  
 favoriteString.clear();  
 AppDatabase db = AppDatabase.*getDbInstance*(FavoriteListActivity.this.getApplicationContext());  
 trackFavourites = db.trackDao().getAllTrackIds();  
 for(int i=0;i<trackFavourites.size();i++){  
 favoriteString.add(trackFavourites.get(i).trackId);  
 }  
 }  
  
 public void getFavourite(String param){  
 checkExpireToken();  
 mService.getFavourite(param, tokenObj.getToken());  
 tracks = jsonService.extractFavourite(mService.getResponseString());  
 }  
  
 public void getNewToken(){  
 mService.getToken();  
 tokenObj=jsonService.extractToken(mService.getResponseString());  
 }  
  
 public void checkExpireToken(){  
 long now = System.*currentTimeMillis*();  
 if ((now - tokenObj.getTimeGetToken()) > tokenObj.getExpireIn() || tokenObj.isEmpty()) {  
 getNewToken();  
 }  
 }  
}

FavouriteAdapter.java

package com.example.spotifyapp;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.spotifyapp.Data.Track;  
  
import java.text.DecimalFormat;  
import java.util.ArrayList;  
import java.util.concurrent.TimeUnit;  
  
public class FavouriteAdapter extends RecyclerView.Adapter<FavouriteAdapter.MyViewHolder> {  
 ArrayList<Track> tracks=new ArrayList<>();  
 Context context;  
 private OnItemClickListener listener;  
  
 public interface OnItemClickListener {  
 void onItemClicked(Track track);  
 void playButtonClicked(Track track);  
 void stopButtonClicked(Track track);  
 void addFavoriteClicked(Track track);  
 }  
  
 public void setOnItemClickListener(OnItemClickListener listener){  
 this.listener =listener;  
 }  
  
 public FavouriteAdapter(Context context, ArrayList<Track> tracks) {  
 this.context= context;  
 this.tracks=tracks;  
 }  
  
 @NonNull  
 @Override  
 public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 LayoutInflater inflater = LayoutInflater.from(context);  
 View view = inflater.inflate(R.layout.track\_favourite\_item, parent, false);  
 return new MyViewHolder(view, listener);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {  
 DecimalFormat df = new DecimalFormat("##");  
 String temp = tracks.get(position).getTrackNumber()+" "+  
 tracks.get(position).getTrackName()+" - "+(TimeUnit.MILLISECONDS.toMinutes(tracks.get(position).getTrackDuration()))+  
 ":"+String.format("%02d", (TimeUnit.MILLISECONDS.toSeconds(tracks.get(position).getTrackDuration())/60));  
 holder.trackDetail.setText(temp);  
 }  
  
 @Override  
 public int getItemCount() {  
 return tracks.size();  
 }  
  
 public class MyViewHolder extends RecyclerView.ViewHolder{  
 TextView trackDetail;  
 Button playButton, stopButton;  
 ImageView addFavorite;  
  
 public MyViewHolder(@NonNull View itemView, final OnItemClickListener listener) {  
 super(itemView);  
 trackDetail = itemView.findViewById(R.id.trackDetail);  
 playButton = itemView.findViewById(R.id.playButton);  
 stopButton = itemView.findViewById(R.id.stopButton);  
 addFavorite = itemView.findViewById(R.id.addFavorite);  
  
 itemView.setOnClickListener((v)->{  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.NO\_POSITION){  
 listener.onItemClicked(tracks.get(position));  
 }  
 }  
 });  
  
 playButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.NO\_POSITION){  
 listener.playButtonClicked(tracks.get(position));  
 }  
 }  
 }  
 });  
  
 stopButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.NO\_POSITION){  
 listener.stopButtonClicked(tracks.get(position));  
 }  
 }  
 }  
 });  
  
 addFavorite.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(listener != null){  
 int position = getAdapterPosition();  
 if(position!=RecyclerView.NO\_POSITION){  
 listener.addFavoriteClicked(tracks.get(position));  
 }  
 }  
 }  
 });  
 }  
  
  
 }  
}

MainActivity.java

package com.example.spotifyapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.widget.Toolbar;  
  
import com.example.spotifyapp.Data.Album;  
import com.example.spotifyapp.Data.Token;  
import com.example.spotifyapp.Service.JsonService;  
import com.example.spotifyapp.Service.MusicDataService;  
  
import java.util.ArrayList;  
import java.util.concurrent.CountDownLatch;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
import java.util.concurrent.TimeUnit;  
  
public class MainActivity extends AppCompatActivity {  
 private RecyclerView mRecyclerView;  
 private ArrayList<Album> albums = new ArrayList<>();  
 private JsonService jsonService = new JsonService();  
 private MusicDataService mService = new MusicDataService();  
 private Token tokenObj = new Token();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 getNewRelease();  
 mRecyclerView = findViewById(R.id.recycleView);  
 GridLayoutManager gridLayoutManager = new GridLayoutManager(this, 2, GridLayoutManager.VERTICAL, false);  
 mRecyclerView.setLayoutManager(gridLayoutManager);  
 mRecyclerView.setHasFixedSize(true);  
 MainAdapter mainAdapter = new MainAdapter(this, albums, new MainAdapter.OnItemClickListener(){  
 @Override  
 public void onItemClicked(Album album) {  
 navigateToAlbum(album);  
 }  
 });  
 mRecyclerView.setAdapter(mainAdapter);  
 mRecyclerView.setLayoutManager(new LinearLayoutManager(this));  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.menu, menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 if(item.getItemId()!=0){  
 navigateToFavorite();  
 return true;  
 }else{  
 return super.onOptionsItemSelected(item);  
 }  
 }  
  
 public void navigateToFavorite(){  
 Intent myIntent = new Intent(this, FavoriteListActivity.class);  
 Bundle bundle = new Bundle();  
 myIntent.putExtras(bundle);  
 startActivity(myIntent);  
 }  
  
 public void navigateToAlbum(Album album){  
 Intent myIntent = new Intent(this, AlbumActivity.class);  
 Bundle bundle = new Bundle();  
 bundle.putString("albumId", album.getAlbumId());  
 myIntent.putExtras(bundle);  
 startActivity(myIntent);  
 }  
  
 public void getNewRelease(){  
 checkExpireToken();  
 mService.getNewRelease(tokenObj.getToken());  
 albums = jsonService.extractNewRelease(mService.getResponseString());  
 }  
  
 public void getNewToken(){  
 mService.getToken();  
 tokenObj=jsonService.extractToken(mService.getResponseString());  
 }  
  
 public void checkExpireToken(){  
 long now = System.currentTimeMillis();  
 if ((now - tokenObj.getTimeGetToken()) > tokenObj.getExpireIn() || tokenObj.isEmpty()) {  
 getNewToken();  
 }  
 }  
  
}

MainAdapter.java

package com.example.spotifyapp;  
  
import android.content.Context;  
import android.graphics.Bitmap;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.GridLayoutManager;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.example.spotifyapp.Data.Album;  
  
import java.util.ArrayList;  
  
public class MainAdapter extends RecyclerView.Adapter<MainAdapter.MyViewHolder> {  
 ArrayList<Album> albums=new ArrayList<>();  
 Context context;  
 private RecyclerView.RecycledViewPool viewPool = new RecyclerView.RecycledViewPool();  
 private RecyclerView mRecyclerView;  
 private final OnItemClickListener listener;  
  
 public interface OnItemClickListener {  
 void onItemClicked(Album album);  
 }  
  
  
 public MainAdapter(Context ct, ArrayList<Album> albums, OnItemClickListener onItemClickListener) {  
 this.context = ct;  
 this.albums = albums;  
 this.listener = onItemClickListener;  
 }  
  
 public class MyViewHolder extends RecyclerView.ViewHolder {  
 TextView itemName, releaseDate, trackCount;  
 ImageView image;  
  
 public MyViewHolder(@NonNull View itemView) {  
 super(itemView);  
 itemName = itemView.findViewById(R.id.*itemName*);  
 releaseDate = itemView.findViewById(R.id.*releaseDate*);  
 trackCount = itemView.findViewById(R.id.*trackCount*);  
 image = itemView.findViewById(R.id.*imageView*);  
 mRecyclerView = itemView.findViewById(R.id.*recycleViewArtist*);  
 }  
 }  
  
 @NonNull  
 @Override  
 public MainAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 LayoutInflater inflater = LayoutInflater.*from*(context);  
 View view = inflater.inflate(R.layout.*grid\_item*, parent, false);  
 return new MyViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull MainAdapter.MyViewHolder holder, int position) {  
 holder.itemName.setText(albums.get(position).getAlbumName());  
 holder.releaseDate.setText(albums.get(position).getReleaseDate());  
 holder.trackCount.setText(String.*valueOf*(albums.get(position).getTotalTrack()));  
 Bitmap bitmap=albums.get(position).getImageBitmap();  
 holder.image.setImageBitmap(bitmap);  
  
 GridLayoutManager gridLayoutManager = new GridLayoutManager(context, 2, GridLayoutManager.*VERTICAL*, false);  
 mRecyclerView.setLayoutManager(gridLayoutManager);  
 mRecyclerView.setHasFixedSize(true);  
 ArtistAdapter artistAdapter = new ArtistAdapter(context, albums.get(position).getArtistNames());  
 mRecyclerView.setAdapter(artistAdapter);  
 mRecyclerView.setLayoutManager(new LinearLayoutManager(context));  
  
 holder.itemView.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v) {  
 listener.onItemClicked(albums.get(position));  
 }  
 });  
 }  
  
 @Override  
 public int getItemCount() {  
 return albums.size();  
 }  
}

================== XML FILE ===================

Activity\_album.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".AlbumActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/black"  
 android:orientation="vertical">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:background="#7B1FA2"  
 android:layout\_weight="0.75">  
 <LinearLayout  
 android:layout\_width="50dp"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:background="#B38857"  
 android:layout\_weight="0.35">  
 <ImageView  
 android:id="@+id/albumImage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 />  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="50dp"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:background="#AB8089"  
 android:layout\_weight="0.65">  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="12dp"  
 android:text="Record lable: "  
 android:textColor="@color/white"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/albumName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="12dp"  
 android:text="Name"  
 android:textColor="@color/white"  
 app:layout\_constraintBottom\_toBottomOf="@+id/textView2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toEndOf="@+id/textView2"  
 app:layout\_constraintTop\_toTopOf="@+id/textView2" />  
  
 <TextView  
 android:id="@+id/textView4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:text="Release Date: "  
 android:textColor="@color/white"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="@+id/textView2"  
 app:layout\_constraintStart\_toStartOf="@+id/textView2"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2" />  
  
 <TextView  
 android:id="@+id/releaseDate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="2021-12-10"  
 android:textColor="@color/white"  
 app:layout\_constraintBottom\_toBottomOf="@+id/textView4"  
 app:layout\_constraintStart\_toStartOf="@+id/albumName"  
 app:layout\_constraintTop\_toTopOf="@+id/textView4" />  
  
 <TextView  
 android:id="@+id/textView7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:text="Track: "  
 android:textColor="@color/white"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="@+id/textView4"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4" />  
  
 <TextView  
 android:id="@+id/trackCount"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="7"  
 android:textColor="@color/white"  
 app:layout\_constraintBottom\_toBottomOf="@+id/textView7"  
 app:layout\_constraintStart\_toStartOf="@+id/releaseDate"  
 app:layout\_constraintTop\_toTopOf="@+id/textView7" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </LinearLayout>  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:background="#FFFFFF"  
 android:gravity="center"  
 android:layout\_weight="0.25">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/trackRecycleView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center|top"/>  
 </LinearLayout>  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Activity\_favourite\_list.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".FavoriteListActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="#FFFFFF"  
 android:gravity="center|top"  
 android:layout\_weight="0.25">  
 <TextView  
 android:id="@+id/status"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 android:text="Empty List"  
 />  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/favoriteListRecycleView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center|top"/>  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text=""  
 android:gravity="center"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/guideline3"  
 app:layout\_constraintVertical\_bias="0.022" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recycleView"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/guidelineEnd"  
 app:layout\_constraintStart\_toStartOf="@+id/guidelineStart"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView" />  
  
 <androidx.constraintlayout.widget.Guideline  
 android:id="@+id/guideline3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 app:layout\_constraintGuide\_percent="0.02" />  
  
 <androidx.constraintlayout.widget.Guideline  
 android:id="@+id/guidelineEnd"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintGuide\_percent="0.96" />  
  
 <androidx.constraintlayout.widget.Guideline  
 android:id="@+id/guidelineDivide"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintGuide\_percent="0.5" />  
  
 <androidx.constraintlayout.widget.Guideline  
 android:id="@+id/guidelineStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintGuide\_percent="0.04" />  
  
 <androidx.constraintlayout.widget.Guideline  
 android:id="@+id/guideline2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 app:layout\_constraintGuide\_percent="0.7" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Artist\_item

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/artistName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/round"  
 android:padding="10dp"  
 android:text="Name"  
 android:textColor="#E6FFFF"  
 android:textSize="10dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

Grid\_item

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="20dp"  
 android:layout\_marginVertical="20dp">  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="220dp"  
 android:layout\_height="match\_parent"  
 android:elevation="10dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" >  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#424242"  
 android:gravity="center|top">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:id="@+id/itemName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:padding="10dp"  
 android:text="Item Name"  
 android:textColor="#FFFFFF"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="200dp"  
 android:layout\_margin="10dp"  
 android:padding="10dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/itemName" />  
  
 <RelativeLayout  
 android:id="@+id/textLayout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:id="@+id/text1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:padding="10dp"  
 android:text="Release Date: "  
 android:textColor="#FFFFFF"  
 android:textStyle="bold"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/releaseDate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="2021-12-10"  
 android:textColor="#FFFFFF"  
 app:layout\_constraintBottom\_toBottomOf="@+id/text1"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toEndOf="@+id/text1"  
 app:layout\_constraintTop\_toTopOf="@+id/text1"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="Tracks: "  
 android:textColor="#FFFFFF"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="@+id/text1"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/text1" />  
  
 <TextView  
 android:id="@+id/trackCount"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="7"  
 android:textColor="#FFFFFF"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toEndOf="@+id/text1"  
 app:layout\_constraintTop\_toTopOf="@+id/textView3" />  
  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
  
 </RelativeLayout>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recycleViewArtist"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textLayout"  
 app:layout\_constraintVertical\_bias="0.0">  
  
 </androidx.recyclerview.widget.RecyclerView>  
  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </RelativeLayout>  
  
 </androidx.cardview.widget.CardView>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Track\_favourite\_item

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/round"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:gravity="center"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:id="@+id/trackDetail"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:text="1 Man in the mirror - 3:59"  
 android:textColor="@color/white"  
 android:textSize="15dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <Button  
 android:id="@+id/playButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="34dp"  
 android:layout\_marginTop="8dp"  
 android:backgroundTint="#00ADB5"  
 android:text="Play"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/trackDetail" />  
  
 <Button  
 android:id="@+id/stopButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="34dp"  
 android:backgroundTint="#00ADB5"  
 android:text="Stop"  
 app:layout\_constraintBottom\_toBottomOf="@+id/playButton"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toEndOf="@+id/playButton"  
 app:layout\_constraintTop\_toTopOf="@+id/playButton"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <ImageView  
 android:id="@+id/addFavorite"  
 android:layout\_width="29dp"  
 android:layout\_height="25dp"  
 android:src="@drawable/ic\_baseline\_delete\_24"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/stopButton"  
 app:layout\_constraintStart\_toEndOf="@+id/playButton"  
 app:layout\_constraintTop\_toTopOf="@+id/playButton"  
 app:tint="#0D1117" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

Track\_item

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/round"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:gravity="center"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:id="@+id/trackDetail"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:text="1 Man in the mirror - 3:59"  
 android:textColor="@color/white"  
 android:textSize="15dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <Button  
 android:id="@+id/playButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="34dp"  
 android:layout\_marginTop="8dp"  
 android:backgroundTint="#00ADB5"  
 android:text="Play"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/trackDetail" />  
  
 <Button  
 android:id="@+id/stopButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="34dp"  
 android:backgroundTint="#00ADB5"  
 android:text="Stop"  
 app:layout\_constraintBottom\_toBottomOf="@+id/playButton"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toEndOf="@+id/playButton"  
 app:layout\_constraintTop\_toTopOf="@+id/playButton"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <ImageView  
 android:id="@+id/addFavorite"  
 android:layout\_width="29dp"  
 android:layout\_height="25dp"  
 android:src="@drawable/ic\_baseline\_favorite"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/stopButton"  
 app:layout\_constraintStart\_toEndOf="@+id/playButton"  
 app:layout\_constraintTop\_toTopOf="@+id/playButton"  
 app:tint="#EE4F4F" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>